CS2010 PS3 - Hospital Tour v4

Released: Sunday, 07 September 2014, 8pm
Due: Monday, 22 September 2014, 8am (extension due to Quiz 1)

Collaboration Policy. You are encouraged to work with other students or teaching staffs (inside or outside this module) on solving this problem set. However, you **must** write Java code **by yourself**. In addition, when you write your Java code, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). This list may include certain posts in CS2010 Facebook group. If you have access to your seniors' CS2010 files (that is, problem sets version 1/2/3), please refrain from looking at their code verbatim. Any deviation from this policy will be considered as cheating. If the offender is caught beyond reasonable doubt, he/she will be punished severely, including referral to the NUS Board of Discipline. It is not worth it to cheat just to get 19% when you will lose out in the other 81%.

2011 Story. In PS2, Steven already mentioned about "attending birth classes" on June 2011 to prepare for the arrival of his baby. One of the event conducted during one of the birth classes is the "hospital tour" where several parents-to-be[1] are gathered together and shown various facilities to help birth in that hospital. The rooms shown are typically the:

- *First class* single room for the new mother and her family to stay for a few nights after delivery,
- Delivery suites (already mentioned in PS2),
- Nursery (a room full of cute newborn babies where the babies are monitored, bathed, fed using formula milk—if the baby's mother choose not to breastfeed the baby, etc),
- Neo-natal Intensive Care Unit (a room that you hope your baby will *never* use),
- Etc

Steven observed that during the tour that he and Grace joined, we were only shown "the better rooms" of the hospital although the hospital itself is much bigger than that. For example, the double and four-bedded rooms, the surgery room (for Caesarean section), the cashier room (where you have to fork out thousand of dollars to pay your bills – especially if you are not a Singapore Citizen → no baby bonus), birth registration room, etc, are not shown for 'logical reasons'. "This is interesting.", Steven thought. As a Computer Scientist, his mind started wandering on what happened behind the scenes so that his hospital tour was impressive...

[1] 99% of the attendees are young married couples expecting their *first* baby – including Steven and his wife Grace that year (2011). Couples do not normally go to birth classes again for their second (or third, etc) baby. This is very true. We did not repeat this process anymore for our second baby (2014).

The Actual Problem. This is a (fictional) story on what happened behind the scenes, many years before Steven visited that hospital...

Assume that you are the manager of your hospital. You know the layout of your hospital (a connected weighted graph) and you have given rating score to each room (the weight of each vertex of your graph). You want to decorate certain room(s) in your hospital so that visitors will feel better when visiting your hospital. As your budget is limited, you want to decorate only an *important* room with the lowest rating score. You define important room as a room that links different buildings in your hospital such that if that room (to be precise, the corridor beside that room) is blocked, the buildings in the hospital becomes 'disconnected'. You want to know the lowest rating score of an important room in your hospital.

For example, suppose your hospital is a connected weighted graph as shown below (room/vertex number is written inside the circles and the rating score of each room is written outside the circles):

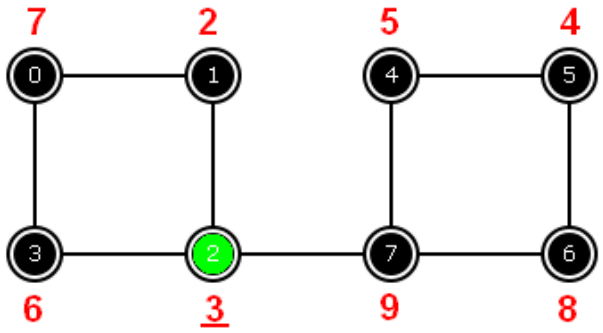

Figure 1: A Sample Hospital Layout

There are two major buildings (`0,1,2,3` and `4,5,6,7`) linked by a corridor `2-7` besides the two important rooms (`2` and `7`). For example, if room `2` is blocked, then people currently in room `0`, `1`, or `3` will not be able to visit people in rooms `4`, `5`, `6`, or `7`. Or in another word, the hospital becomes 'disconnected'. Similar situation if room `7` is blocked. The other rooms are not important rooms. For example, if room `1` is blocked, people from room `0` can still go to any other rooms in the hospital (other than room `1` of course) via path `0-3-2` and so on.

Now, among the two important rooms `2` versus `7`, room `2` has lower rating score (3 points) than room `7` (9 points). So, you will decorate room `2`. You will report 3 as the answer of your own query: "What is the lowest rating score of an important room in your hospital?".

The skeleton program `HospitalTour.java` is already written for you, you just need to implement one (or more) method(s)/function(s):

- `int Query()`
  Query your Graph data structure (already implemented in `HospitalTour.java` and returns the lowest weight (lowest rating score) of an important vertex (important room) of your graph (your hospital) (these rating scores are stored in another Vector of Integers). We guarantee that the input graph is a connected undirected unweighted graph and all rating scores are positive integers not larger than 100000. If your hospital has no important room, you will not decorate any room, i.e. just return -1. Note that the answer for this query is still unique even though there can be more than one important rooms with similar lowest rating score.

- If needed, you can write additional helper methods/functions to simplify your code.

Subtask A (50 points). The given hospital layout map is a small weighted tree ($1 \leq V \leq 50$). Example:

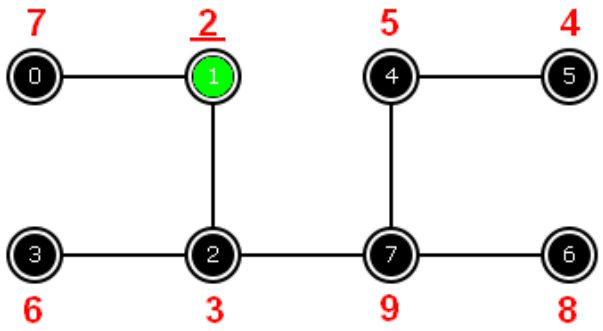

Figure 2: A Simplified Hospital Layout (Tree)

There are four important rooms in this hospital: `1`, `2`, `4`, `7`. Among these four, important room `1` has the lowest rating score: 2. Therefore, the answer for your own query is 2.
PS: It is possible to solve Subtask A right after studying Lecture 05 (without Lecture 06).
Students are encouraged to implement solution for Subtask A as early as possible.

Subtask B (Additional 25 points). The given hospital layout map is a small, connected, and weighted general graph ($1 \leq V \leq 50$). Use `Sample.txt` to see if your program can pass Subtask B.
PS: We have more rigorous test cases to judge your final submission. If you receive Wrong Answer here, discuss with others about the possible corner cases. The expected solution for Subtask B requires Lecture 06 material.

Subtask C (Additional 25 points). Same as Subtask B, but the graph is larger ($1 \leq V \leq 1000$).
PS 1: We have even more rigorous test cases than Subtask B to judge your final submission.
PS 2: For this Subtask C, efficiency matters. The expected solution is $O(V^2 + VE)$ per test case.
PS 3: Our test data has $E = O(V)$, i.e. $E = c \times V$ for a small constant factor c.
PS 4: It is quite likely that your first submission is either Wrong Answer or Time Limit Exceeded for Subtask C due to the increasing level of strictness. Therefore, do not wait until the last day to attempt Subtask C.
PS 5: If you are unable to pass Subtask B, you are unable to pass Subtask C+D either as both use Subtask B test data and much more.

Optional Subtask D (No point at all). Your program must be able to solve Subtask C and also must be *extremely efficient* as the graph is very large ($1 \leq V + E \leq 100000$). Hint: This requires a special algorithm beyond CS2010 and previously used for CS2010R (it is dropped in 2014 due to a certain situation in 2013). Beware that this Subtask D (which carries 0 point) can take hours to complete if you choose to attempt it.
PS 1: A code that solves Subtask D should be able to solve Subtask A, B, and C easily.
PS 2: If you are unable to pass Subtask C, do not bother submitting to Subtask D.
PS 3: The expected solution is $O(V + E)$ per test case.
PS 4: If you encounter stack overflow, try running your Java program with: '-Xss8m' flag.

Note: The official test data has been uploaded to Mooshak online judge, but it is hidden from your view. The time limit setting in Mooshak online judge for Subtask A, B, C, and D (the old R-option) are all 1 second (i.e. rather strict). You are encouraged to generate and post additional test data in Facebook group. Note that `HospitalTourVerifier.java` is available but not distributed as it contains partial solution to this PS. Please post your custom-made test data but only after you verify its correctness, i.e. check for these conditions before uploading them:

1. The rating scores must be positive integers not larger than 100000.
2. The input graph must be connected and undirected.
3. Make sure the size of the graph is within the constraints stipulated in each subtask.